Federating Digital Contact Tracing using Structured Overlay Networks

Silvia Ghilezan, Simona Kašterović, Luigi Liquori, Bojan Marinković, Zoran Ognjanović, Tamara Stefanović

Federating Digital Contact Tracing using Structured Overlay Networks

Silvia Ghilezan [1, 2], Simona Kašterović [2], Luigi Liquori [4], Bojan Marinković [3], Zoran Ognjanovć [1], and Tamara Stefanović [2]

[1]Mathematical Institute of the Serbian Academy of Sciences and Arts, Belgrade, Serbia
[2]Faculty of Technical Sciences, University of Novi Sad, Serbia
[3]Clarivate, Serbia
[4]INRIA & Université Côte d'Azur, France

Abstract

In this paper we present a comprehensive, yet simple, extension to the existing systems used for Digital Contact Tracing in Covid-19 pandemic. The extension, called *BubbleAntiCovid19* (BAC19), enables those systems, regardless of their underlying protocol, to enhance their sets of traced contacts and to improve the global fight against pandemic during the phase of opening boarders and enabling more traveling. BAC19 is a structured overlay network, or better, a Federation of mathematical Distributed Hash Tables. Its model is inspired by the Chord and Synapse structured overlay networks. The paper presents the architecture of the Overlay Network Federation and shows that the federation can be used as a formal model of Forward Contact Tracing.

Keywords: Covid-19, Digital Contact Tracing, Distributed Hash Tables, Structured Overlay Networks, Bluetooth, GPS

1 Introduction

One of the biggest challenges of today is to slow down the spreading of SARS-CoV-2 virus producing Covid-19 pandemic; *Prevention, Testing and Tracing* are the main pillars of the solution. Contact Tracing of an infected person is essential to control the spread of the disease.

Tracing. Contact tracing is the process of identifying, notifying, and monitoring people who came in close contact with an individual who was tested positive for an infectious disease, like Covid-19, while he/she was infectious.

Contact tracing benefits the fight with the pandemic at multiple levels. Identifying and quarantining close contacts limits their ability to spread the disease. Therefore, in a period in which the disease and its effects are still being investigated, contact tracing plays a key role in preventing the further spread of the disease. Furthermore, contact tracing data helps medical experts to find the origin of the virus and learn more about the nature of the virus.

Manual Contact Tracing. Contact tracing has mostly been done manually since many centuries ago just by taking note on a simple piece of paper the list of persons and goods you get in contacts with (see e.g. *La Peste* by A. Camus [3]). In the actual days, manual contact tracing could be exploited using simple telephone calls. Identifying contacts is done through an interview with the person infected with the virus. Each person is then contacted by phone. Health Authorities should quickly alert people who are close contacts that they may have been exposed to the virus. The sooner the contacts are notified, the lower the risk of the spreading further. However, due to the highly contagious nature of the SARS-CoV-2 virus and the fact that symptoms can manifest after many days (or even never, e.g. *asymptomatic cases*), manual contact tracing does not give satisfactory results. Health departments and authorities do not have enough employees to do manual contact tracing. It must be further emphasized that the SARS-CoV-2 can be transmitted not only by direct contact, but also by indirect contact. The reason is that infected people can leave virus droplets on any physical object they touch. In this case, manual contact tracing is ineffective. For the reasons stated above, digital contact tracing has been considered already at the beginning of the Covid-19 pandemic.

1.1 Problem

There is a plethora of digital contract tracing applications in use all over the world fighting the Covid-19 pandemic [8, 23]. They are developed on very different paradigms, centralized [4] vs. decentralized [7], GPS based (very few indeed because of a clear violation of privacy) vs. Bluetooth Low Energy based (the majority). The rush to make these applications work in the shortest time led to their great diversity. The most important open problem is their interoperability. There are many ongoing efforts to make a federation of these different systems. Herein, we address this problem and propose a solution based on mathematical models of overlay networks.

1.2 Contributions

We develop a formal federation overlay network, called *BubbleAntiCovid19* (BAC19), for connecting different digital contact tracing applications, which are currently in use all over the world. The model is based on the well-known model of Structured Overlay Network protocols like e.g. Chord [27, 28], Kademlia [24], Synapse [18]. We prove that BAC19 provides a complete and fully exhaustive retrieving procedure of people that get in touch with other people having tested positive to the Covid-19 disease. Hence, BAC19 is proven to be a simple yet

powerful *interconnection* of already existing digital contact tracing applications that - by construction - do not communicate with each others as such providing their efficient interoperability.

As far as we know, the mathematical model and techniques presented in this paper have not been considered in other approaches.

1.3 Overlay networks in a nutshell.

Structured Overlay Networks are suitable models of scalable and efficient organization of resources on the Internet. They represent logical organizations, independent on underlying network infrastructure that physically connects available assets. Overlay networks have been proven as very resilient tool in the situation when some parts of the underlying infrastructure fail or become overloaded or corrupted.

1.4 Organization of the paper

The rest of the paper is organized as follows. Section 2 presents classifications of digital contact tracing applications. Section 3 reviews Chord and Synapse protocols of overlay networks. Section 4 briefly reviews basic notions of Abstract State Machines and some related work by the authors. Section 5 introduces the BAC19 system and proves the completeness and full exhaustiveness of the retrieving procedure. Section 6 presents a discussion on other proposals for providing interoperable frameworks for digital contact tracing. Section 7 concludes the paper. Appendix give an overview of different digital contact tracing applications that are in current use against the pandemic.

2 Digital Contact Tracing Applications

Advances in digital technology have enabled smartphones and other digital devices to be used for contract tracing. Particularly, more and more countries are showing interest in digital contact tracing applications (DCT apps) implemented for smartphones. Despite the great variety among these applications, all contact tracing apps work on the principle of automatic data exchange with nearby devices. When a user of a particular contact tracing app is identified as infected, a special report is uploaded to the DCT app server. Based on that report, close contacts of the infected person (also DCT app users) are informed that they have been in a contact with a positive user and/or the app calculates their exposure risk. The identity of the infected persons is not disclosed in order to protect their privacy. Existing contact tracing apps can be classified based on two criteria:

- Contact-tracing technology;
- System architecture.

More information about the classification of the existing contact tracing apps can be found in [26, 29].

2.1 Classification by contact-tracing technology

In order for two people to be in close contact, they need to stay in the same place, at a short distance, for a long enough period of time. Therefore, the main data type used by the contact tracing apps is location data. There are various technologies for collecting and tracking location data, and contact tracing apps can be divided into two major categories depending on whether they track absolute or relative location:

- Absolute location apps – Apps that track the absolute location of their users are mostly based on GPS technology. Location data is stored in the form of geolocation coordinate pair. These apps are also known as Geolocation-based DCT apps.
- Relative location apps – Apps that track relative location of their users are mostly based on Bluetooth technology. These apps are also known as proximity DCT apps. A boarding pass or a ticket for a specific event can also be considered as relative location data. In order to use this kind of data, some contact tracing apps deploy QR code technology.

2.2 Classification by system architecture

Since the data is collected from users, their processing should be addressed. When it comes to DCT app data managing, the responsibility can be on a central authority or on each user individually. Therefore, contact tracing apps can be divided into three major categories depending on the architecture of the underlying system:

- Centralized apps - data are solely managed by a central server;
- Hybrid apps - multiple nodes can manage the data, but the control is centralized;
- Decentralized apps - each user is managing his/her data.

In centralized apps, the central server is responsible for ID generation, risk analysis and notifications. In decentralized apps these functionalities are moved to the user devices and the central server is only an encounter point. The hybrid architecture proposes decentralized ID generation and centralized risk analysis. There are a few proposed hybrid contact tracing protocols, but their implementation in real contact tracing apps is still waiting. More about these protocols can be found in [1]. For that reason, we will focus on apps with centralized and decentralized architecture, and we will provide an overview of the existing contact tracing apps based on the above classifications in Appendix.The results are summarized in Figure 1, which is motivated by [26].

2.3 Geolocation-based apps.

Geolocation-based DCT apps record past geo-trajectories of every user, and the calculation of exposure risk of a user is based on the intersection of its past trajectories and trajectories of patients. We give a brief review of the existing Geolocation-based apps in Appendix A.1.

Two main advantages of geolocation-based DCT apps are the following:

1) Geolocation-based DCT apps are compatible with manual contact tracing. Compatibility of geolocation-based DCT apps and manual contact tracing has mutual benefits. On the one hand, past geo-trajectories of a patient can be added to an app by the contact tracer even if the patient did not use the app. This enables the app to warn more users. On the other hand, the app can give the information about places with higher exposure risk to a contact tracer, so that the contact tracer can identify high-risk service workers.

2) Another advantage is that geolocation-based DCT apps can recognize patterns of disease's spreading and locations with higher exposure risk, and they can inform health authorities about it.

Nevertheless, geolocation-based DCT apps have also disadvantages. The major challenges are privacy concerns, which cause low adoption rate of these applications. User's privacy can be violated in several ways. Recording all user's trajectories can result in revealing user's personal information such as identity, home address, work address, the identity of the patient and revealing user's exposure risk to other users. These problems have been elaborated in more details in [5].

2.4 Bluetooth-based apps

Bluetooth-based DCT apps record direct contacts of the users. A device generates a unique, randomized identifier and assigns it to a user. There are two kinds of identifiers: static, identifiers do not change over time, and dynamic, identifiers change over time. During a direct contact devices exchange identifiers and save received identifiers. Once a user is identified as positive in the application, other users can calculate their exposure risk by checking whether they received a patient's identifier. We give a brief review of the existing Bluetooth-based apps in Appendix A.2.

Depending on whether the exposure risk is calculated by the central server or the user's device, we have centralized and decentralized apps, respectively, see Figure 2, which is motivated by [16].

Centralized apps raise privacy concerns and questions about massive surveillance. People often do not trust servers and as a consequence there is a low adoption rate of these applications. On the other hand, the advantage of centralized apps is the possibility for health authority to make a transmission graph and learn more about the virus. Also, the possibility of false positive users is reduced.

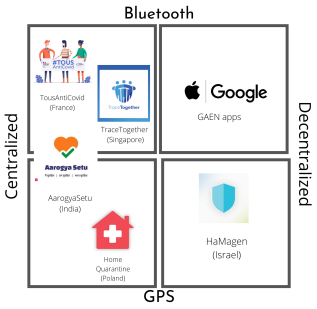

Figure 1: Classification of analyzed applications

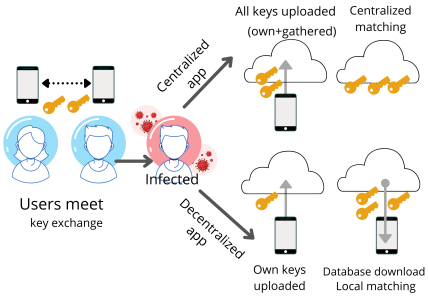

Figure 2: Centralized vs Decentralized Bluetooth-based apps

As we have already observed, privacy issues lead to low adoption rate and decrease the efficiency of the application. In order to solve the problem of distrust of the central server, decentralized apps were designed. However, decentralized methods also raise some privacy concerns, for example the identity of a patient can be easily revealed.

The major disadvantage is that Bluetooth-based DCT apps work only if both users have installed the *same* application.

In order to take advantage of both types of these apps, Bluetooth-GPS apps were designed, see Appendix A.3. Given the different characteristics of DCT apps, the question arises whether it is possible to aggregate their data in order to track contacts more effectively. The answer can be found in overlay networks.

3 Overlay Networks

Overlay networks are the way to organize available assets, as mentioned in Section 1.

Some overlay networks are implemented in a form of Distributed-Hash-Tables (DHTs). One of DHT protocols is the Chord protocol. It was introduced in [27, 28]. Nodes that are part of a Chord system form a ring shaped

network. The basic operations of a Chord node are entering and leaving the system and the mapping given key onto the corresponding node of the system using consistent hashing.

The correctness and efficiency of the Chord's protocol lookup procedure was in the focus of several papers, e.g. [27, 28, 20, 22]. However, these properties will not be in the focus of this paper. Our goal is to deliver information of every affected node, so we will not use any presented improvements to speed up the process of getting results, but to linearly pass every node in a Chord network, to be sure that no information is missed.

Interconnection of several overlay networks is a very hard problem since different networks may use different protocols, and even in the case of several DHT networks that use the same protocol (e.g. Chord) it is enough that every overlay network uses *its own hash function* and information between two of them cannot be exchanged. A proposal to solve this issue was given by defining the Synapse protocol in [18]. Its performances were analyzed in [19], whereas one real-life proof of concept was developed in [21]. For the purpose of this paper we will consider the so-called, *white-box* version of the Synapse protocol that, in short, allows to consider all the keys as they were using the same hash table (see [18] for details). Again, since we will use the linear search procedure in one Chord network we can be sure that information will be retrieved if it exists in the system.

4 Abstract State Machines and Chord

In this section, we briefly review basic notions of Abstract State Machines.

Abstract State Machine (ASM) [2, 14] is a formalization method to model algorithms at the appropriate abstraction level. An ASM $\mathcal{A}$ is defined as a program $Prog$ which consists of:

- an at most countable set of states, its subset of initial states, and
- a finite number of transition rules,

where states are first order structures over a fixed signature, whereas transitional rules:

- update (:=),
- sequential (*seq* ... *endseq*),
- conditional (*if* ... *then* ... *else* ... *endif*),
- parallel (*par* ... *endpar*),
- nondeterministic (*choose* $v \in U$ *satisfying* $g(v)$ ... *endchoose*) and
- universal (*forall* v *with* g ... *endforall*)

represent next-state functions. An execution of one of the last two types of rules introduces a variable v. In the case of nondeterministic rule, the transition is executed with a value of v which satisfies a guard g, while in the case of universal rule, the transition is executed simultaneously for all values v which satisfy a guard g. An ASM can interact with its environment using external functions (oracles) by providing arguments to oracles and receiving the corresponding results.

In a distributed case with many agents, every agent executes its own program and has its own partial view of a global state. The nullary function Me allows an agent to identify itself among other agents. The global program is the union of all agents' programs, whereas a transition between two states is obtained by an evaluation of transition functions of all agents.

An ASM $\mathcal{A}$ models a real system $\mathcal{S}$ in terms of evolution of states described by runs. A run of $\mathcal{A}$ is a (in)finite sequence of S_0, S_1, S_2, ... where S_0 is an initial state, and every S_{i+1} is obtained from S_i by executing a transitional rule. In this paper we consider only the runs in which states are global and agents' moves are atomic (instantaneous). The most general kind of runs for a distributed ASM are partially ordered runs. To prove properties of partially ordered runs, thanks to the results proved in [14], the attention may be restricted to their linearizations that are sequential runs and satisfy the following fundamental properties:

- All linearizations of the same finite initial segment of a run have the same final state.
- A property holds in every reachable state of a run iff it holds in every reachable state of any of its linearization.

Note that this implies that it is enough to find only one sequence of transitions and the runs that are considered here and start from the same initial state will have the same final state.

The key notions introduced in [20, Definition 5.1] are:

- stable states in a Chord network, where a state of a network is stable if the successor (predecessor) pointers of all nodes form an ordered ring, and
- regular runs, where a run is regular if it is a linearization such that nodes leave and enter the network only in stable states.

Having these notions, the paper [20] proves that the presented formalization of Chord consistently maintains the topological structure of rings and manipulates with distributes keys. In Section 5 we will explain that our model of BAC19 satisfies the mentioned constraints and that results from [20] hold also for BAC19.

On the other hand the papers [19, 18] recognize the fact that the search procedure of the Synapse protocol is not complete and fully exhaustive. This is due to the fact that the lookup procedure of the Chord network can skip some of the synapse nodes, and thus not to spread the query to all networks that

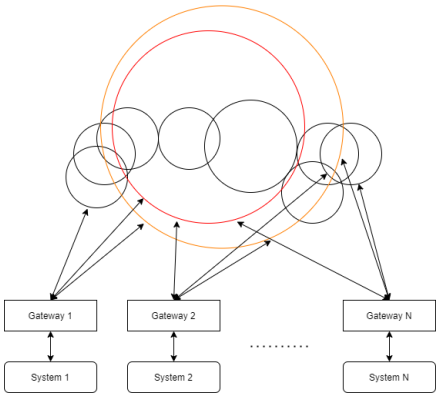

Figure 3: BAC19 Federation Overlay Network

are reachable. To avoid this situation we redefine the lookup procedure with Algorithm 1, not to skip any node.

5 System BAC19

In this section, we propose the design of the system *BubbleAntiCovid19* (BAC19), which is a formal federation overlay network for connecting different digital contact tracing applications that are currently in use all over the world.

BAC19 federation (Figure 3) consists of several Chord networks:

- a network for each person/device of his/her first contacts (black circles in Figure 3),
- dedicated *red* network to connect all infected persons (red circle in Figure 3),
- dedicated *amber* network to connect all the first contacts of infected persons (amber circle in Figure 3),

and

- *Gateways* (black rectangles in Figure 3) as the connections to the existing digital contact tracing systems (black rounded corners rectangles in Figure 3).

The first connection between the proposed extension and an existing system for contact tracing is called *Gateway*. The purpose of a *Gateway* is to maintain communication between two parts and to transform messages in a way that both sides can communicate efficiently.

The most important thing is to maintain the mappings between identifiers (IDs) used on both sides of a *Gateway*. As we could see in Section 2, some systems periodically change IDs, so the possibility to trace those changes is

FindSuccessor =
For Given key
$//successor(id(Me))$ is responsible for key
if $member\_of(key, id(Me), successor(id(Me)))$ **then**
  Respond With $successor(id(Me))$
else
  //Me forwards query to its successor
  Forward Query To $successor(id(Me))$
end

Algorithm 1: FindSuccessor

vital for functioning of BAC19. Regarding IDs, our goal is to have one identifier per one person/device regardless of how many systems it appears in. We argue that this is possible to achieve. First, it is possible to use sufficiently large codomain of the hash function (e.g. 2^{128}). Also, it is possible to select enough parameters of a person/device so that it can be uniquely identified. We are not storing any other attribute of a person/device except a newly introduced identifier in our extension.

More precisely, with respect to the specifications that are provided in [20, 19] we need to introduce the following changes:

- the set
$$Network = \{red, amber, net_1, \ldots, net_N\}, N \in \mathbb{N}$$
to denote all possible networks, where N is the number of possible persons/devices in the proposed extension;

- the set $Time$ and the function
$$contact\_time() : (Chord \cup \{amber\}) \times Chord \to Time$$
to denote the time of the contact between two persons;

- the external function $current\_date()$ to get the current date.

To obtain *completeness* and *full exhaustiveness* of the retrieval procedure the rule FindSuccessor, which finds a responsible node for a given ID, is changed as in the way presented in Algorithm 1.

With this proposal we are not compromising performances of the extension by much. Since the number of contacts of a person is relatively small, it is manageable to allow increasing the complexity of the worst case retrieval from $O(logN)$ to $O(N)$. In the predefined time-slots our extension will receive the following information from a system:

- all identified infected cases since the last import (Algorithm 2),

- all confirmed cases that are not infected anymore since the last import (Algorithm 3),

For all $inf \in NewCases$
  Invoke PUT Of Network red To Store inf
For all $id \in net_{inf}$
  if $contacttime(amber, id) < t$ *or* $contacttime(amber, id) = undef$
 then
 | Set $contacttime(amber, id) = t$
end

Algorithm 2: Put

For all $inf \in Healed$
  Invoke LEAVE Of Network red for inf

Algorithm 3: Leave

- all identified contacts since the last import in the form of the tuple $\langle id_i, id_j, t \rangle$ with the meaning that persons id_i and id_j had a risk contact at t timestamp. For the purpose of providing privacy protection timestamp should be kept at the precision of days. Unfortunately, this type of communication is not possible with the systems that are categorized as decentralized Bluetooth systems, since the fact that contact tracing computation is performed at users' devices and not shared with the central storage. These systems can only share newly identified cases and their time of recovery (see Algorithm 4).

par
  Set $contacttime(id_i, id_j) = t$
  Set $contacttime(id_j, id_i) = t$
endpar

Algorithm 4: Set contact time

Also, if needed it is possible to introduce the new *Gateway* with the purpose to enter manually recognized contacts to the system.

When information is received from origin systems, as the first step BAC19 will connect all newly recognized infected cases to the *red* network, as well as to remove all cured. A node will remain in the *red* network until its recovering is confirmed. All IDs that are recognized as the risk contacts of a person/device (e.g. id_i) will be added to its bubble. They will stay there until $t + 14$ *days*, where t is the time of their contact. If the id_i is the member of the *red* network all the members of its network will be added to the *amber* network and stay there during the same time frame $t + 14$ *days*. If a contact is already in the *amber* network timestamp will be updated to the higher value (Algorithm 5).

During the opposite way of communication, BAC19 will pass on information to all nodes in the *amber* network to *Gateways*. If an identifier is recognized in the set of mappings for the particular origin system, the corresponding information is transferred to the origin system to alert (if not already) the person/device

For all $net_{id_i} \in Network \backslash \{red\}$
  For all $id_j \in net_{id_i}$
    if $contacttime(id_i, id_j) + 14\ days > currentdate()$ **then**
| Invoke LEAVE Of Network id_j for id_i
end

Algorithm 5: Leave of network

that she/it had risk contact with an infected person at stored timestamp. Also, BAC19 is capable to send information on the second level contacts (the result is stored in the set *Result*, Algorithm 6):

seq
  Invoke GET all nodes from *amber* and store the result in *Amber*
  For all $id \in Amber$
    Invoke GET all nodes from net_{id} and append the result to *Result*
endseq

Algorithm 6: Get all nodes

Namely, for all nodes of the *amber* network it is possible to go through every origin bubble and pass those identifiers to the *Gateways*. Then the origin systems can inform those persons that they should increase their awareness since they are second level contacts.

Using the results from [20, 19] we prove the following statement:

Theorem 1. *The proposed extension stores and retrieves only up-to-date information on Covid-19 positive cases (identified by the origin systems) and their contacts and makes it available to all origin systems.*

Proof. It is shown in the paper [20] that it might happen that stable states in a Chord network cannot be achieved if Leave and/or Put rules are fired in unstable states. Thus, to avoid that in BAC19, it is necessary to ensure that the executions of each of Algorithm 2 - Algorithm 6 do not intertwine.

Executions of the proposed extension are performed in the controlled environment. Due to the scheduled time intervals for running different tasks, the nodes' leaving from the bubbles will not happen during the unstable states, i.e., there will be only runs compatible with conditions of [20, Theorems 5.3, 5.4 and 5.7]. Also, the fact that the rule FINDSUCCESSOR is changed guarantees that all nodes will be contacted during the search procedure, and that the retrieving procedure of the Synapse protocol [19] is complete and fully exhaustive.

As a consequence of the mentioned adaptations of the proposed Chord model, all possible execution of BAC19 fulfill conditions from [20, Theorems 5.3, 5.4 and 5.7]. So, with these modifications starting from the given state BAC19 will always reach the stable state, and the retrieved information will be up-to-date and valid. □

6 Discussion

The paper [30] proposes building a common API. This approach is rather similar to the extension proposed in this paper. However, these approaches have also two significant differences:

- while [30] building API connection points between each of two different origin systems that are connected, our extension proposes a version to common bus where each of the origin systems communicates with the proposed extension and in this way reduces and simplifies the number of connection points that needs to be maintained when several origin systems are connected;
- with BAC19 we are simplifying also information that is being exchanged, and we do not violate privacy in the origin systems (since our extension does not collect information of origin DCT system).

ETSI GS-E4P presents in [9] an interoperability framework for pandemic contact tracing systems which allows the centralized and decentralized modes of operation to fully interoperate.

A guideline on Interoperability specifications for cross-border transmission chains between approved apps by the European Community [6] proposes a Federation Gateway Service for synchronizing the diagnosis keys (keys of infected users) across backend servers of each national app. However, this approach focuses only on Google/Apple exposure notification apps because the majority of European countries have developed this kind of apps, and also because one Google/Apple exposure notification app can detect the contact with a user of another Google/Apple exposure notification app. In this paper we do not focus on a certain type of DCT apps, we want to achieve the connection between them regardless the contact-tracing technology and their system architecture. We leave to the the reader to envisage the following scenario:

- Alice lives in the region which has centralized DCT *System A*, while Bob lives in the region which has centralized DCT *System B*. Bob has spent some time in the region A, and both of them are traveling together side by side with negative RT-PCR tests. However, Bob developed symptoms of Covid-19 after couple of days and was confirmed as positive.

If *System A* and *System B* are part of BAC19, it would be enough that only one of Alice and Bob had installed system from the other region just in the time of travel for Alice to be informed that she is the first contact of an infected person.

7 Conclusions

In this paper we have presented BAC19 a new and efficient overlay network connecting existing systems for digital contact tracing. The advantages of BAC19 (its usage) are:

- a person does not install anything new on his/her mobile device (except a new application which is used in the region that this person is visiting);
- the overlay does not store any personal sensitive information;
- the overlay is independent regarding how the origin system calculated contacts or is it based on Bluetooth or GPS technology;
- the overlay supports manual entry of recognized contacts;
- there are no new highly complicated calculations of possible contacts beside those that are performed by the original contact tracing systems.

The presented extension BAC19 is the so-called forward tracing system (finding all contacts of an infected person). We plan to explore the possibilities to adapt BAC19 to also enable backward tracing (finding the source of infection using contacts).

A DCT apps - overview

A.1 Geolocation-based DCT apps

Home Quarantine. At the beginning of the Covid-19 pandemic, Ministry of Digital Affairs of Poland developed the Home Quarantine app [25]. This is a typical example of a centralized app which deploys GPS technology. It is developed to support the authorities, especially the police and social services, with adequate information about people undergoing mandatory home quarantine. Users are also required to upload their digital photos. So, aside the GPS technology the app also uses face recognition. The app is mandatory for anyone who has developed coronavirus symptoms. It should be emphasized that Poland also developed the ProteGO Safe app for alerting users of close contact with an infected person based on The (Google/Apple) Exposure Notification (GAEN) system.

The Shield (HaMagen). In March 2020, Israeli Ministry of Health developed The Shield app [17]. This is a typical example of a decentralized app which deploys GPS technology. Location data is stored in the phone. If a user tests positive, he/she can upload his/her location history to the central server. Once the user uploaded his/her location history, it is added into a JSON file that is updated with new data on an hourly basis. Matching the locations happens on the phone. If the match is found, the app shows you the exact time and location. The app is later updated to work with Bluetooth technology but on a voluntary basis, every user can choose whether to use the proximity data or not.

A.2 Bluetooth-based DCT apps

Blue-Trace protocol apps. Singapore's Government Technology Agency in collaboration with Ministry of Health in March 2020 released the TraceTogether

app [13] that allows digital contact tracing using the custom BlueTrace protocol. Australia has later adopted the protocol and released the CovidSafe app [10]. Contact tracing is done using Bluetooth Low Energy and proximity data is encrypted and stored only on the users phone. Users in the contact log are identified using anonymous time-shifting "temporary IDs". If a user tests positive for the infection, the Ministry of Health requests his/her contact log. The user has the right to choose whether to share the contact log or not. If the user chooses to share the log, the contact log is uploaded to a central server and the health authority is then responsible for matching the log to contact detail and informing close contacts of the infected user. These apps are examples of Bluetooth-based centralized apps. It should also be noted that Singapore solved the problem of tracing people who don't use smartphones by enabling the app to work with Token - a physical Bluetooth-based device.

ROBERT protocol app. The French National Assembly released the StopCovid app in May 2020. The app has later been renamed to TousAntiCovid [11]. It allows digital contact tracing using the ROBust and privacy-presERving proximity Tracing protocol (ROBERT protocol). It also deploys Bluetooth technology and belongs to the category of centralized apps. The difference between this app and apps based on the BlueTrace protocol relates to confirmation of positive users. More precisely, in France when a person is confirmed to be positive, the lab gives a patient a QR code and the scanned code is the proof for the app that you are infected. It is up to you to share this information with the app, and if you choose to share this information with a central server, the server is responsible for alerting your close contacts.

Google/Apple exposure notification apps. In April 2020 Google and Apple announced the joint work on decentralized Bluetooth-based protocol named The (Google/Apple) Exposure Notification (GAEN) system [15]. Many states then developed different apps using the Google/Apple Exposure Notification framework including Austria (Stopp Corona app), Germany (Corona-Warn-App), Italy (Immuni), Canada (COVID Alert) etc. The principle by which applications work is as follows. During a close contact, user's phones exchange random Bluetooth identifiers. These identifiers change frequently and the information about exchanged ID's is stored on the user's phone. When a user gets infected, he/she can decide to upload ID's he/she was using the last 14 days to the server. Phones of all users periodically download the list of ID's which belong to the infected users and does the matching locally.

A.3 Bluetooth-GPS apps

Apps that deploy both Bluetooth and GPS technology are rare. One app of this kind is the *Aarogya Setu app* [12], developed by National Informatics Centre that comes under the Ministry of Electronics and Information Technology, Government of India. Aarogya Setu is following the centralized approach, and is one of the world's fastest growing applications. The app mainly uses proximity data and GPS data are recorded only once in 30 minutes. The location data is mainly used to identify the locations where you might have caught the infection

and identify potential hotspots that may be developing when multiple infected people visit the same place. Interaction between users is recorded by exchange of Device Identification Numbers (DiD's) which are static. Contact tracing data is kept on the phone. Council of Medical Research (ICMR) shares the list of Covid-19 positive persons with the Aarogya Setu server, and information about contact tracing is uploaded to the server only if you are tested positive. The central server is then responsible for alerting your close contacts.

Funding. This work was partly supported by: the Science Fund Republic of Serbia #6526707 AI4TrustBC.

References

[1] Nadeem Ahmed, Regio A. Michelin, Wanli Xue, Sushmita Ruj, Robert Malaney, Salil S. Kanhere, Aruna Seneviratne, Wen Hu, Helge Janicke, and Sanjay K. Jha. A survey of covid-19 contact tracing apps. *IEEE Access*, 8:134577–134601, 2020.

[2] Egon Börger and Robert F. Stärk. *Abstract State Machines. A Method for High-Level System Design and Analysis*. Springer, 2003.

[3] Albert Camus. *La peste*. Gallimard, 1947.

[4] Claude Castelluccia, Nataliia Bielova, Antoine Boutet, Mathieu Cunche, Cédric Lauradoux, Daniel Le Métayer, and Vincent Roca. ROBERT (RObust and privacy-presERving proximity Tracing protocol). Technical report, Inria and Fraunhofer AISEC, 2020.

[5] Xiang Cheng, Hanchao Yang, Archanaa S. Krishnan, Patrick Schaumont, and Yaling Yang. KHOVID: interoperable privacy preserving digital contact tracing. *CoRR*, abs/2012.09375, 2020.

[6] eHealth Network. Interoperability specifications for cross-border transmission chains between approved apps, 2020.

[7] Carmela Troncoso et al. Decentralized Privacy-Preserving Proximity Tracing. Technical report, École Polytechnique Fédérale de Lausanne, ETH Zurich, KU Leuven, Delft University of Technology, University College London, Helmholtz Centre for Information Security, University of Torino, ISI Foundation, 2020.

[8] ETSI. Comparison of existing pandemic contact tracing systems. Technical Report DGS E4P-002, 2021. Work in progress.

[9] ETSI. Pandemic proximity tracing systems: Interoperability framework. Technical Report DGS E4P-007, 2021. v1.0.1 draft.

[10] Government of Australia. CovidSafe. `https://www.covidsafe.gov.au/` Accessed January 9, 2021.

[11] Government of France. TousAntiCovid. `https://gitlab.inria.fr/stopcovid19` Accessed January 10, 2021.

[12] Government of India. Aarogya Setu. `https://aarogyasetu.gov.in/` Accessed January 10, 2021.

[13] Government of Singapore. TraceTogether. `https://www.tracetogether.gov.sg/` Accessed January 9, 2021.

[14] Yuri Gurevich. Evolving algebras 1993: Lipari guide. In Egon Börger, editor, *Specification and validation methods*, pages 9–36. Oxford University Press, 1993.

[15] Jaap-Henk Hoepman. A Critique of the Google Apple Exposure Notification (GAEN) Framework. *ArXiv*, abs/2012.05097, 2020.

[16] Jianwei Huang, Vinod Yegneswaran, Phillip Porras, and Guofei Gu. On the privacy and integrity risks of contact-tracing applications, 2020.

[17] Israel Ministry of Health. HaMagen. `https://govextra.gov.il/ministry-of-health/hamagen-app/download-en/` Accessed January 9, 2021.

[18] Luigi Liquori, Cédric Tedeschi, Laurent Vanni, Francesco Bongiovanni, Vincenzo Ciancaglini, and Bojan Marinkovic. Synapse: A scalable protocol for interconnecting heterogeneous overlay networks. In Mark Crovella, Laura Marie Feeney, Dan Rubenstein, and S. V. Raghavan, editors, *NETWORKING 2010, 9th International IFIP TC 6 Networking Conference, Chennai, India, May 11-15, 2010. Proceedings*, volume 6091 of *Lecture Notes in Computer Science*, pages 67–82. Springer, 2010.

[19] Bojan Marinković, Vincenzo Ciancaglini, Zoran Ognjanović, Paola Glavan, Luigi Liquori, and Petar Maksimović. Analyzing the exhaustiveness of the synapse protocol. *Peer Peer Netw. Appl.*, 8(5):793–806, 2015.

[20] Bojan Marinković, Paola Glavan, and Zoran Ognjanović. Proving properties of the chord protocol using the ASM formalism. *Theor. Comput. Sci.*, 756:64–93, 2019.

[21] Bojan Marinković, Luigi Liquori, Vincenzo Ciancaglini, and Zoran Ognjanović. A distributed catalog for digitized cultural heritage. In Marjan Gusev and Pece Mitrevski, editors, *ICT Innovations 2010 - Second International Conference, ICT Innovations 2010, Ohrid, Macedonia, September 12-15, 2010. Revised Selected Papers*, volume 83 of *Communications in Computer and Information Science*, pages 176–186, 2010.

[22] Bojan Marinković, Zoran Ognjanović, Paola Glavan, Anton Kos, and Anton Umek. Correctness of the chord protocol. *Comput. Sci. Inf. Syst.*, 17(1):141–160, 2020.

[23] Tania Martin, Georgios Karopoulos, José L. Hernández-Ramos, Georgios Kambourakis, and Igor Nai Fovino. Demystifying COVID-19 Digital Contact Tracing: A Survey on Frameworks and Mobile Apps. *Wireless Communications and Mobile Computing*, 2020(8851429):29, 2020.

[24] Petar Maymounkov and David Mazières. Kademlia: A peer-to-peer information system based on the XOR metric. In Peter Druschel, M. Frans Kaashoek, and Antony I. T. Rowstron, editors, *Peer-to-Peer Systems, First International Workshop, IPTPS 2002, Cambridge, MA, USA, March 7-8, 2002, Revised Papers*, volume 2429 of *Lecture Notes in Computer Science*, pages 53–65. Springer, 2002.

[25] Ministry of Foreign Affairs Republic of Poland. Home Quarantine. `https://www.gov.pl/web/diplomacy/home-quarantine-monitoring-by-taketask` Accessed January 10, 2021.

[26] Patrick Ocheja, Yang Cao, Shiyao Ding, and Masatoshi Yoshikawa. Quantifying the privacy-utility trade-offs in COVID-19 contact tracing apps, 2020.

[27] Ion Stoica, Robert Tappan Morris, David R. Karger, M. Frans Kaashoek, and Hari Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In Rene L. Cruz and George Varghese, editors, *Proceedings of the ACM SIGCOMM 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, August 27-31, 2001, San Diego, CA, USA*, pages 149–160. ACM, 2001.

[28] Ion Stoica, Robert Tappan Morris, David Liben-Nowell, David R. Karger, M. Frans Kaashoek, Frank Dabek, and Hari Balakrishnan. Chord: a scalable peer-to-peer lookup protocol for internet applications. *IEEE/ACM Trans. Netw.*, 11(1):17–32, 2003.

[29] Qiang Tang. Privacy-preserving contact tracing: current solutions and open questions, 2020.

[30] Marko Vukolic. On the interoperability of decentralized exposure notification systems. *CoRR*, abs/2006.13087, 2020.